

FIG. 1

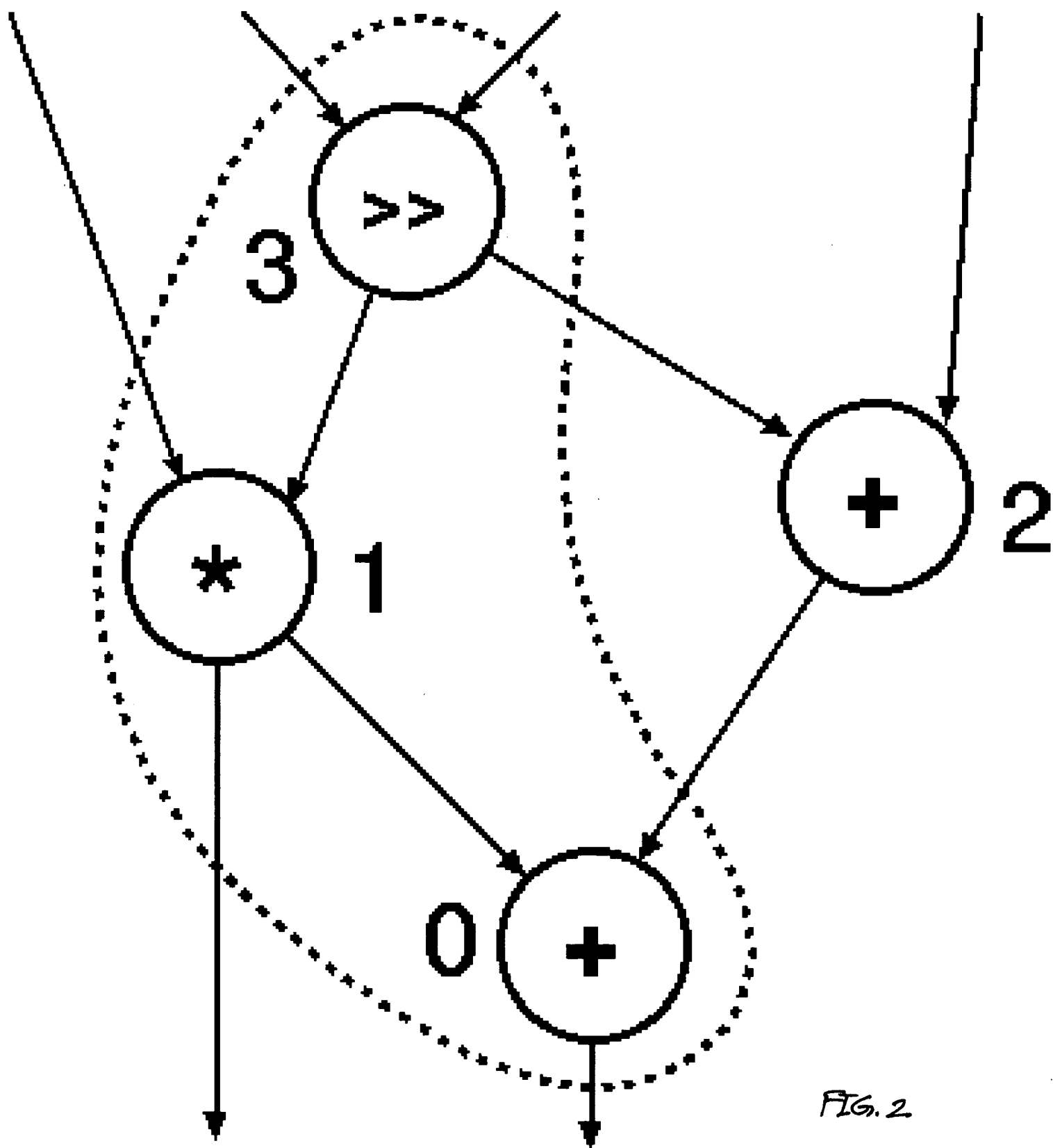


FIG. 2

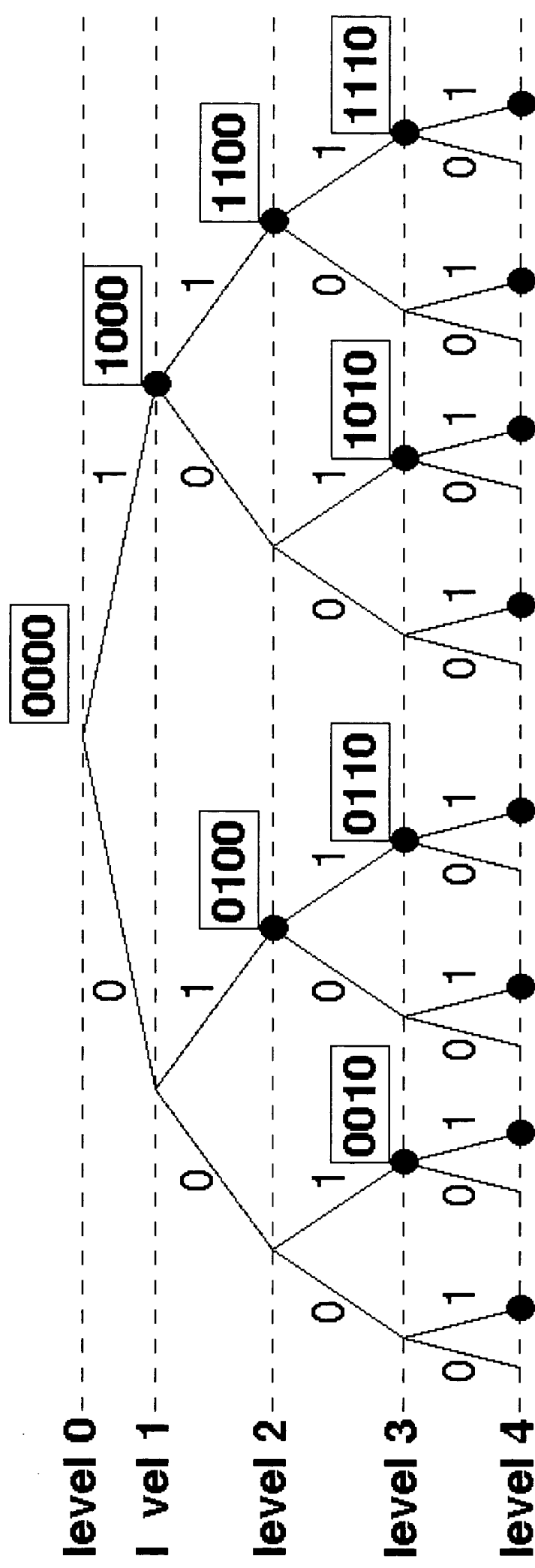


FIG. 3

```

identification() {
  for (i = 0; i < NODES; i++) cut[i] = 0;
  topological_sort();
  search(1, 0);
  search(0, 0); }

search(current_choice, current_index) {
  cut[current_index] = current_choice;
  if (current_choice == 1) {
    if (!output_port_check()) return;
    if (!convexity_check()) return;
    if (input_port_check()) {
      calculate_speedup();
      update_best_solution(); } }
  if ((current_index + 1) == NODES) return;
  current_index = current_index + 1;
  search(1, current_index);
  search(0, current_index); }

```

Figure 4: The identification algorithm.

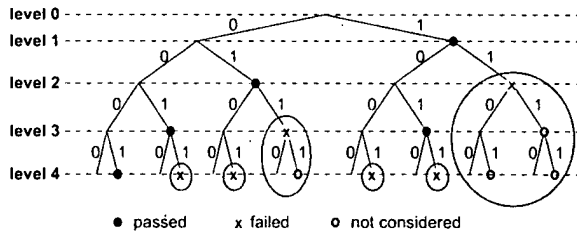


Figure 5: The execution trace of the algorithm for the graph given in Figure 2 and $N_{out} = 1$.

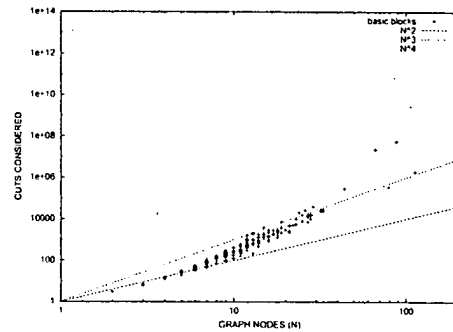


Figure 6: Number of cuts considered by the algorithm with $N_{out} = 2$ and any N_{in} .

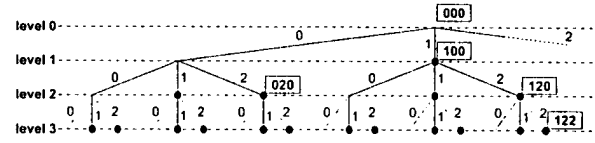
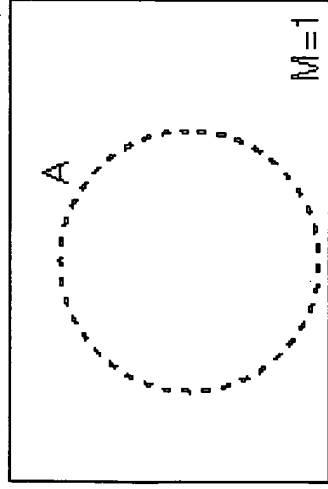
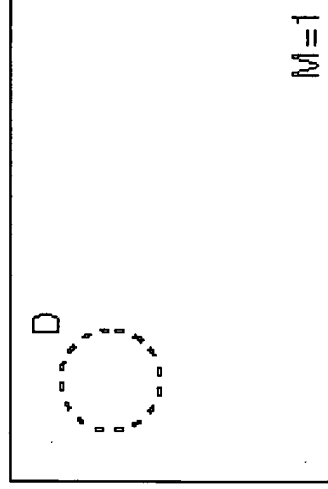


Figure 7: A search tree for two cuts.

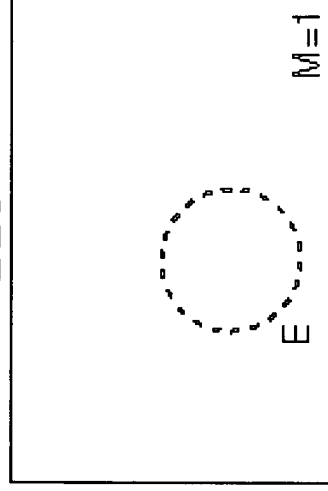
BB1



BB2



BB3



Iteration 1

$A > D$ and $A > E$

Iteration 2

$E > B + C - A$ and $E > D$

Iteration 3

$F + G - E > B + C - A$ and
 $F + G - E > D$

Solution

Fig. 8

